Constraint-Preconditioned Inexact Newton Method for Hydraulic Simulation of Large-Scale Water Distribution Networks

Edo Abraham, *Member, IEEE*, and Ivan Stoianov

***Abstract*—Many sequential mathematical optimization methods and simulation-based heuristics for optimal control and design of water distribution networks rely on a large number of hydraulic simulations. In this paper, we propose an efficient inexact subspace Newton method for hydraulic analysis of water distribution networks. By using sparse and well-conditioned fundamental null space bases, we solve the nonlinear system of hydraulic equations in a lower-dimensional kernel space of the network incidence matrix. In the inexact framework, the Newton steps are determined by solving the Newton equations only approximately using an iterative linear solver. Since large water network models are inherently badly scaled, Jacobian regularization is employed to improve the condition number of these linear systems and guarantee positive definiteness. After presenting a convergence analysis of the regularized inexact Newton method, we use the conjugate-gradient (CG) method to solve the sparse reduced Newton linear systems. Since CG is not effective without good preconditioners, we propose tailored constraint preconditioners that are computationally cheap because they are based only on invariant properties of the null-space linear systems and do not change with flows and pressures. The preconditioners are shown to improve the distribution of eigenvalues of the linear systems and so enable a more efficient use of the CG solver. Since contiguous Newton iterates can have similar solutions, each CG call is warm-started with the solution for a previous Newton iterate to accelerate its convergence rate. Operational network models are used to show the efficacy of the proposed preconditioners and the warm-starting strategy in reducing computational effort.**

***Index Terms*—Constraint preconditioners, hydraulic analysis, inexact Newton method, null-space algorithm, preconditioned conjugate gradient, water distribution networks.**

I. Introduction

WATER distribution networks (WDNs) are typically part of an aging infrastructure, which face challenges to efficiently and sufficiently serve a growing population under more stringent economic and environmental constraints. Some of the main operational challenges for WDNs include reducing pressure-driven leakage (or high levels of nonrevenue water) [1], monitoring and control of drinking-water quality [2], and reducing the cost associated with high-energy usage in pumping [3]. In recent years, novel valve technologies and advances in sensor and control capabilities, and considerable growth in the provision of cloud-based analytics and IT solutions are enabling these operational problems to be solved in near real time and for increasingly larger systems [4].

For a given network topology with specific physical components and their properties (i.e., pipe length and diameters, material properties such as pipe flow resistance, node elevations, control settings of valves and pumps), the hydraulic analysis problem involves the computations of flows in pipes and pressures at the nodes. We solve a set of nonlinear equations that govern the conservation of mass (flow continuity) and conservation of energy principles to simulate the system behavior. The network hydraulic analysis problem is, however, more than just a simulation tool since it plays a vital role in optimal control, resilience analysis, and design problems for WDNs. Many of the WDN optimization problems are difficult because they often involve a large number of nonconvex (and some nonsmooth) constraints, and a mixture of continuous and integer decision variables; the resulting large-scale optimization problems are either nonconvex nonlinear programs or mixed-integer nonlinear programs [5]. The use of simulation-based heuristics in solving difficult optimization problems in water-resources management has been an active area of research for at least two decades; a large list of formulations can be found in the review paper [6]. For example, by coupling conventional hydraulic simulation software, such as EPANET [7] with evolutionary algorithms, a least cost design/rehabilitation of a water network is carried out in [8].

Scalable mathematical optimization approaches have also been proposed based on hydraulic analysis. In [3], extended time hydraulic simulations are used within a sequential linear programming (SLP) framework to find optimal pump schedules that reduce the cost of electricity by shifting pumping to cheap tariff periods. An adjoint approach is employed in [9] to optimize reservoir dynamics over a finite horizon, where hydraulic analysis is used to compute the system state, adjoint equations, and, subsequently, the objective function and Jacobians of constraints. In [2], in addition to minimizing energy cost or pump operational time, a discrete-time optimal scheduling problem is also reformulated to minimize deviation of chemical concentrations from desired values for water quality. Since their control variables are the only independent variables, and other state variables (i.e., flow, pressure, tank level, and chemical

Manuscript received December 17, 2015; revised March 11, 2016; accepted March 22, 2016. Date of publication April 29, 2016; date of current version September 15, 2017. This work was supported by the NEC-Imperial "Big Data Technologies for Smart Water Networks" project. The authors acknowledge the EPSRC Industrial CASE Studentship project EP/I501444/1, from which the case study models BWFLnet and BWKWnet were derived. Recommended by Associate Editor S. Azuma.

The authors are with the Department of Civil and Environmental Engineering, Imperial College London SW7 2AZ, U.K. (e-mail: edo.abraham04@imperial.ac.uk; ivan.stoianov@imperial.ac.uk).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCNS.2016.2548418

concentration values) implicitly depend on the control variables through the nonlinear hydraulic constraints, a generalized reduced gradient (GRG) method is used in [2]. The direction and step sizes in the control variables are computed using the Jacobian of the constraints [10]; at each step of the GRG nonlinear program solver, the implementation in [2] calls the EPANET hydraulic and water-quality simulator to solve for the system states and numerically approximate the Jacobians of the constraints or to check constraint feasibility for steps taken. Motivated by leakage reduction, we consider in [11] optimal pressure and dynamic topology control of sectorized WDNs using pressure-reducing valves (PRVs). We employ an SLP method, where a sequence of linear programs is solved by linearizing constraints at each iterate. We use an Armijo-type line search coupled with hydraulic analysis to keep the SLP iterates strictly feasible and guarantee improved convergence properties to local optima. Since the computational bottleneck of all these optimization approaches is the repeated application of hydraulic analysis, this paper is concerned with their efficient solution.

As for many other nonlinear equations, the classical algorithm for solving the hydraulic equations is the Newton method, where a sequence of linear equations is formed and solved at each iteration [12]. Since solving these linear systems to high accuracy is the bottleneck of the Newton method, multiple approaches have been proposed to reduce the computational burden [13]. The Newton method for hydraulic analysis has a Jacobian with the same saddle-point structure that equality-constrained quadratic programs have [13], [14]. In the analysis of water and gas pipe networks, both Schur [15] and null space [13], [16]–[18] algorithms have been adapted to exploit this structure. More recently, null-space methods have been shown to be a more efficient alternative in repeated demand-driven simulations [13], [18].

In this paper, we propose a null-space inexact Newton method for hydraulic analysis. The inexact method solves the nonlinear hydraulic equations to the same level of accuracy as an exact method. In the inexact framework, however, the system of linear equations is solved only approximately to find the Newton steps when the initial iterations are far away from the solution, with the accuracy becoming progressively finer as we get closer to a solution. Iterative linear solvers are employed in inexact Newton methods because, unlike direct solvers, they can be terminated early and so allow a tradeoff between accuracy and computational effort for the linear solvers at each Newton iterate [19], [20].

For WDNs, which are networks with a sparse graph, the Newton steps are computed by solving linear equations projected onto a smaller dimensional kernel space of the flow conservation constraints. By using sparse basis, we obtain reduced linear equations that are sparse, symmetric, and positive definite. The conjugate gradient (CG) [14] method is therefore considered for solving them. Compared to direct linear solvers, which can produce denser factorizations due to fill-in (i.e., the number of nonzeros in the factors of a matrix A but with corresponding zeros in the matrix A itself [21]), iterative methods, such as the CG algorithm, exploit sparsity better because they involve only matrix-vector multiplication. This can result in smaller computational effort along with reduced storage requirements that only depend on the number of nonzero entries of the linear systems [22].

Iterative solvers can, however, have poor performance when the linear systems are ill-conditioned. The linear systems we consider are inherently very badly conditioned due to the large scale of variation in pipe friction coefficients and flows in operational water network models. Therefore, preconditioning techniques are important to accelerate the convergence rate of the CG method [22]. We consider the use of a simple Jacobian regularization technique [23] in the Newton method to prevent the condition numbers from becoming too large or the linear systems becoming semidefinite; we propose appropriate condition number bounds for the regularization that will not negatively affect the convergence properties of the Newton method. We also derive, from the physical properties of the network links, different tailored constraint preconditioners [14] that enhance the rate of convergence of the CG iterations. Iterative methods can have a much improved convergence if warm-started with a good initial guess. Since contiguous Newton iterates have close solutions, warm-starting is exploited by initializing each CG iteration by a solution to a previous Newton iterate.

The remainder of this paper is organized as follows. In the next section, we will present the hydraulic analysis problem, the Newton method for solving it, and a proof of its convergence properties. Section III discusses a null-space Newton algorithm for hydraulic analysis and summarizes its computational advantages relative to a Schur approach. In Section IV, we describe the inexact Newton method for hydraulic equations and consider Jacobian regularizations. Section V considers the use of CG within the inexact method and the new tailored preconditioners. The efficacy of these preconditioners and a warm-starting approach are shown using a numerical study with a number of operational network models, followed by our conclusions in Section VI.

Notation: For a vector $v \in \mathbb{R}^n$, we define the usual p-norms as $\|v\|_p := (\sum_{i=1}^{n} |v_i|^p)^{1/p}$, $p = 1$, 2 and $\|v\|_p = \max_i |v_i|$ if $p = \infty$. $\|v\|_X := \sqrt{v^T X v}$ for a square matrix X. For a matrix A, $\|A\|_p = \max_{\|x\|=1}(\|Ax\|_p/\|x\|_p)$, where $\|Ax\|_p$, $\|x\|_p$ are the corresponding vector norms. A^T denotes the transpose of the matrix A. For an invertible matrix X, we denote its condition number by $\kappa(X)_p := \|X\|_p\|X^{-1}\|_p$. The (right) null space of a matrix A is also denoted by $\ker(A)$.

II. Problem Formulation

In this paper, we consider a demand-driven model for a WDN, where water demand is estimated for consumption nodes from measurements across the system and are assumed known. The network is represented by a graph $\mathscr{G}(V, E)$, where the set of vertices V consists of the set of nodes with unknown pressures, and the set of reservoirs and tanks with known piezometric pressure heads. The set of edges E represents the set of n_p links, which includes pipes and a set of n_v PRVs E_v, where $E_v \subset E$. For a network with n_p links connecting $n_n (< n_p)$ unknown head nodes, and n_0 known head nodes, we define the vector of unknown flows and pressure heads as $q = [q_1, \ldots, q_{n_p}]^T$ and $h = [h_1, \ldots, h_{n_n}]^T$, respectively. Let pipe p_j have flow q_j

going from node i to node k, and with pressure heads h_i and h_k at nodes i and k, respectively. The frictional headloss (or flow resistance) across the pipe is represented as

$$h_i - h_k = r_j |q_j|^{n-1} q_j \tag{1}$$

where r_j, the resistance coefficient of the pipe, can be modelled as either independent of the flow or implicitly dependent on flow q_j and is given as $r_j = \alpha L_j/(C_j^n D_j^m)$. The variables L_j, D_j, and C_j denote the length, diameter, and roughness coefficient of pipe j, respectively. The triplet α, n, and m depend on the energy-loss model used; Hazen—Williams (HW: $r_j = 10.670 L_j/(C_j^{1.852} D_j^{4.871})$) and Darcy—Weisbach (DW) are two commonly used frictional head-loss formulae [23]. In DW models, the formulae vary with Reynolds number and the dependence of the resistance coefficient on flow is implicit; see [24, (1), (2)]. The mass and energy conservation laws that describe the water distribution system are defined by the set of equations

$$A_{11}(q)q + A_{12}h + A_{10}h_0 + A_{13}u = 0 \tag{2a}$$

$$A_{12}^T q - d = 0 \tag{2b}$$

where the variables $h_0 \in \mathbb{R}^{n_0}$ and $d \in \mathbb{R}^{n_n}$ represent the known heads (e.g., at a reservoir or tank) and demands consumed at nodes, respectively, and $u \in \mathbb{R}^{n_v}$ are the control inputs (i.e., variables that enforce additional headloss across their respective PRVs). The elements of the matrix $A_{12} \in \mathbb{R}^{n_p \times n_n}$, which is the node-edge incidence matrix relating the n_p links with the n_n unknown head nodes, are defined as

$$A_{12}(j,i) = \begin{cases} -1, & \text{if pipe j leaves node i} \\ 1, & \text{if pipe j enters node i} \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

where inflows into a node are considered positive and outflows as negative. The incidence matrix A_{10} for the fixed-head nodes is defined using the same convention as (3). Demands d consumed at the nodes are represented by positive values. The diagonal matrix $A_{11} \in \mathbb{R}^{n_p \times n_p}$ has the elements

$$A_{11}(i,i) = r_i |q_i|^{n_i - 1}, \quad i = 1, \ldots, n_p \tag{4}$$

where $r \in \mathbb{R}^{n_p}$ is the vector of frictional resistance factors of the pipes and $n \in \mathbb{R}^{n_p}$ are constants related to the frictional head loss. For the HW and DW friction formula for pipes, $n_i = 1.85$ and $n_i = 2$, respectively. For valves, $n_i = 2$ and an empirical value for r_i is generally supplied by the valve manufacturer.

At each step of the SLP algorithm for optimal pressure control in [11], we solve the nonlinear (2) for given control settings u, known reservoir peizometric heads h_0 and demands d. The Newton method is the classical method for solving nonlinear equations and unconstrained optimization problems because of its fast local convergence properties [12], [25]; it has similarly been applied to solve hydraulic equations as far back as in [26]. Let $f(q,h)$ represent the nonlinear equations in (2). By considering the Jacobian of $f(q,h)$ with respect to the unknown $x := [q\ h]^T$, and using the head-loss model for the ith link (4), the Newton iterations for the solution of (2) solve the linear equations $\nabla f(x^k)(x^{k+1} - x^k) = -f(x^k)$, i.e.,

$$\begin{bmatrix} F^k & A_{12} \\ A_{12}^T & 0 \end{bmatrix} \begin{bmatrix} dq \\ dh \end{bmatrix} = \begin{bmatrix} -\left(A_{11}(q^k)q^k + A_{12}h^k + A_{10}h_0 + A_{13}u^k\right) \\ d - A_{12}^T q^k \end{bmatrix} \tag{5}$$

where $\begin{bmatrix} dq \\ dh \end{bmatrix} = \begin{bmatrix} q^{k+1} - q^k \\ h^{k+1} - h^k \end{bmatrix}$, $N := \mathtt{diag}(n_i)$, $i = 1, \ldots, n_p$, and $F^k := NA_{11}(q^k)$. An equivalent representation for the set of linear (5) is

$$\begin{bmatrix} F^k & A_{12} \\ A_{12}^T & 0 \end{bmatrix} \begin{bmatrix} q^{k+1} \\ h^{k+1} \end{bmatrix} = \begin{bmatrix} c^k \\ d \end{bmatrix} \tag{6}$$

where $c^k := (F^k - G^k)q^k - A_{10}h_0 - A_{13}u$, $G^k := A_{11}(q^k)$, and all of the variables on the right-hand side are known at the kth Newton iteration. In Lemma 1, we first state the convergence properties of the Newton method for hydraulic analysis. This proof allows us to study/guarantee convergence properties of modified null-space algorithms by posing them as inexact Newton methods in Section IV.

Lemma 1: Let $x^* := [q^*\ h^*]^T \in D$, with an open convex set D be a nondegenerate solution of (2), that is, the Jacobian $\nabla f(x^*)$ is not singular, and let $\{x^k\}$ be the sequence of states generated by the Newton iteration (5). For $x^k \in D$ sufficiently near x^*, the Newton sequence exists (i.e., $\nabla f(x^i)$ is nonsingular for all $i > k$) and has local superlinear convergence (i.e., better than a linear rate of convergence).

Proof: ([13, Lemma 1]): We show in [13] that $f(\cdot)$ is continuously differentiable in $\mathbb{R}^{n_p + n_n}$. If we assume x^* is nondegenerate, the proof is a standard result and is given in [12, Theor. 11.2]. ■

Remark 1: It can also be shown that the Jacobian of the loss functions is Lipschitz either when a Darcy—Weisbach equation is used (for laminar/low flows) or when the solution does not have zero flows. In such cases, the Newton algorithm will have local quadratic convergence [12, Theor. 11.2].

III. Null-Space Method for Hydraulic Analysis

The Newton linear (6) has a *saddle-point structure* [27]; if the 2×2 block structure is considered, the A_{11} block is symmetric positive definite or semidefinite, $A_{21} = A_{12}^T \in \mathbb{R}^{n_n \times n_p}$, $n_p \geq n_n$, and $A_{22} = 0$. This structure arises in many equality-constrained optimization problems in fluid mechanics, electrical circuit analysis, computational structural mechanics, and other applications; see [27, Sec. 6] for a large list of literature on such applications and a complete survey of solution methods.

If A_{11} is invertible, considering the block partitions of (6) and applying block substitutions (a Schur complement reduction [27, Sec. 5]), an equivalent linear system with a smaller number of primary unknowns is

$$A_{12}^T (F^k)^{-1} A_{12} h^{k+1} = A_{12}^T (F^k)^{-1} c^k - d \tag{7a}$$

$$q^{k+1} = (F^k)^{-1}(c^k - A_{12}h^{k+1}). \tag{7b}$$

This Schur reformulation in (7) is what is often called the nodal formulation in the hydraulics engineering literature [15] because the pressure heads at the nodes are solved for first and the flows are computed by substitution.

An efficient approach in solving these involves eliminating the linear equality constraints to solve a reduced problem in the null space of the constraints [13], [18]. Suppose that the columns of a full-rank matrix $Z \in \mathbb{R}^{n_n \times n_l}$, $n_l = n_p - n_n$, span the null space of A_{12}^T, that is, $A_{12}^T Z = 0$. Then, any flow q^{k+1} that satisfies the equality constraint (2b) can be written as the sum $q^{k+1} = A_{12}w + Zv$, for some $w \in \mathbb{R}^{n_n}$ and $v \in \mathbb{R}^{n_l}$, where the constant vector $q^* := A_{12}w \in \mathbb{R}^{n_p}$ satisfies the mass balance constraint (2b). Therefore, substituting for q^{k+1} in the first block row of (6) and premultiplying by Z^T gives

$$Z^T F^k Z v = Z^T [c^k - F^k q^*]. \tag{8}$$

With $q^{k+1} = q^* + Zv$ computed, the heads h^{k+1} are calculated by solving

$$A_{12}^T A_{12} h^{k+1} = A_{12}^T (c^k - F^k q^{k+1}). \tag{9}$$

A null-space algorithm-based Newton method first solves for q^* such that $A_{12}^T q^* = d$, and then iteratively solves (8) and (9) in sequence until convergence is achieved. Of course, (9) need not be solved at each iteration but when the iterates are near convergence because the flow equations (8) do not depend on the pressure heads h^k; this reduces computational effort. A discussion on converegnce criteria can be found in [13] and [23].

Additional reasons why null-space methods can be computationally superior compared to Schur approaches for hydraulic simulation are summarized here. Where the null-space dimension n_l is small, the linear system in (8) is smaller than the Schur method linear (7a). Since F^k is diagonal, the null-space problem will be sparse if Z is sparse. With an appropriate choice of Z, the number of nonzeros in $Z^T F^k Z$ is much less than the number of nonzeros in $A_{12}^T F^k A_{12}$ for most WDN models. The matrices $Z^T F^k Z$ can be shown to be symmetric positive definite (SPD) even when F^k is singular if $\ker(F^k) \cap \ker A_{12}^T = \{0\}$. Depending on the method of choice for computing Z, a number of null-space methods can be adopted; our work in [13] contains a study of various approaches for generating very sparse and well-conditioned fundamental null-space basis Z. In addition, the matrix Z is computed only once for multiple hydraulic simulations. Algorithm 1 shows the null-space Newton method tailored to demand-driven hydraulic analysis.

The matrix coefficient of (9), $A_{12}^T A_{12}$, is similarly SPD—see [18, App.] for proof that the incidence matrix for WDNs A_{12} has full rank, and positive definiteness follows. Since this matrix depends only on the network topology and does not change with Newton iterations or demand, a single numeric factorization can be used for multiple hydraulic analyses. In extended time simulations, we need to solve for different q^* as the demands d vary with time. Now, since $q^* = A_{12}w$, $w \in \mathbb{R}^{n_n}$ and substituting for q^* in (2b), we obtain

$$A_{12}^T A_{12} w = d. \tag{10}$$

Therefore, the same single factorization of the SPD matrix $A_{12}^T A_{12}$ can be used to solve for the pressure heads in (9) and for q^* by forward and back substitutions, resulting in further computational savings.

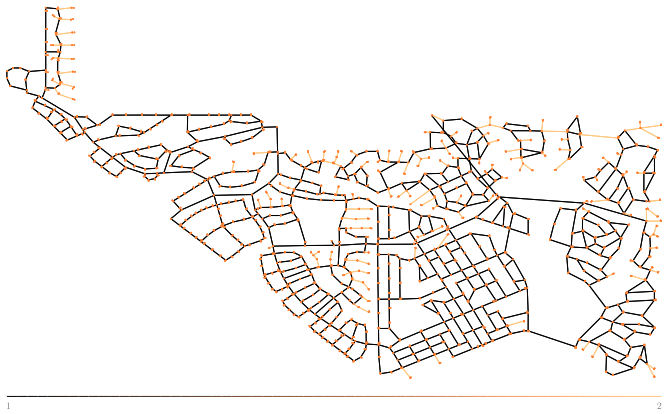


Fig. 1. Proportion of links (≈58%) involved in flow updates are in black edges for network WCnet; see the appendix for network details.

Another computational saving comes from the fact that only a fraction of the network graph edges is projected into the null space when appropriate fundamental null-space basis is used [13]. The repeated headloss computations for both Hazen—Williams and Darcy—Weisbach models take a significant fraction of total flops used by the Newton iterations. For example, for the operational network shown in Fig. 1, only about 60% of pipes are involved in the loop (8) and 40% of links are excluded from unnecessary computations by our approach in [13]. For the sparser network BWFLnet, 70% of links will not be involved in loop equations. At each Newton iteration k, by updating headlosses G^k, F^k, and c^k only for pipes involved in the loop equations, [13] shows a CPU time reduction of up to 1/3 for the sparsest networks.

Algorithm 1 Null Space Newton Method

Preprocessing for extended time simulations: Compute all constants
1) Compute null-space basis Z
2) Factorize $A_{12}^T A_{12}$ (i.e., compute L such that $LL^T = A_{12}^T A_{12}$)
3) Compute and factorize preconditioners

Preprocessing for given demand d:
1) Solve for q^* from (10): $LL^T w = d$, $q^* \leftarrow A_{12}w$

Input: δ_N, $k_{\max}$, (q^*, L, Z), q^1, h^1, u, h_0
set $k = 1$, and compute $G^1, F^1, \|f(q^1, h^1)\|_\infty$
while $\|f(q^k, h^k)\|_\infty > \delta_N$ AND $k \le k_{\max}$ **do**
  $F^k = \text{Regularize}(F^k)$
  Solve $Z^T F^k Z v^k = b^k$
  $q^{k+1} = q^* + Z v^k$
  Recompute G^k, F^k
  IF Near Convergence, THEN
    Solve $LL^T h^{k+1} = b(q^{k+1})$
    Compute the Residual error $\|f(q^k, h^k)\|_\infty$
  Set k to $k + 1$
end while

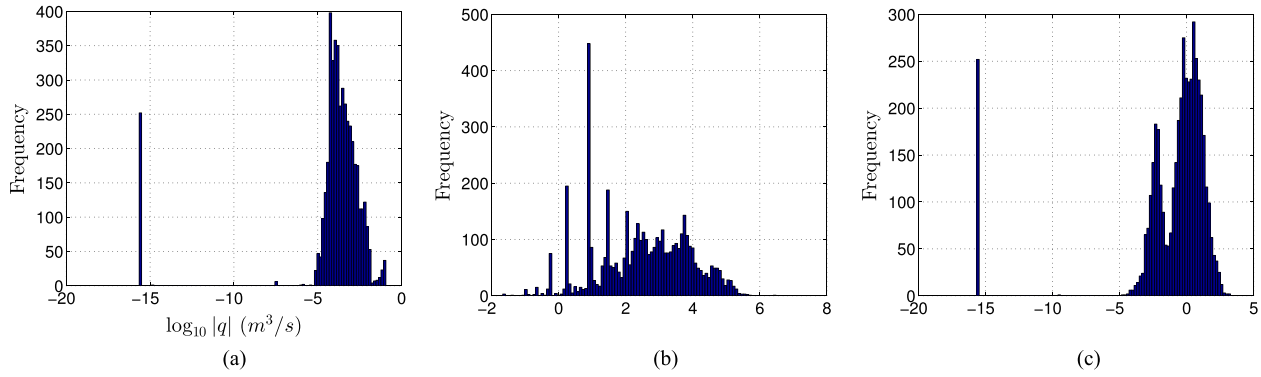


Fig. 2. Histograms showing the distribution of (a) pipe flows for BWKnet network at 8:15 A.M.. (b) frictional loss coefficients for the pipes and (c) associated elements in the diagonal matrix A_{11} in (5). Zero values are plotted as machine precision ($\mathtt{e}ps, \approx 2e^{-16}$) here. (a) Flow magnitudes. (b) r (flow resistance coefficients). (c) $\mathtt{d}iag(A_{11})$.

IV. Inexact Newton Method for Solving Hydraulic Equations

By Lemma 1, Algorithm 1 is a Newton method for the hydraulic equations in (2). At each Newton iteration k, the algorithm solves the linear equation $\nabla f(x^k)s^k = -f(x^k)$ in (5) to find the Newton steps $s^k := x^{k+1} - x^k$. In exact Newton methods, these linear equations are solved exactly to sufficiently small tolerances using direct or iterative solvers; for example, the solutions would satisfy $\|\nabla f(x^k) + f(x^k)\| \leq e_{\text{tol}}\|f(x^k)\|$, where $e_{\text{tol}} > 0$ can be made as small as possible depending on the machine precision and condition number of the problem. Solving these linear systems to high accuracy is the bottleneck of the Newton method [12, Sec. 11.1]. Since exact methods are often computationally expensive, inexact Newton methods solve the linear equations only approximately to find a step s^k that satisfies the milder condition

$$\left\|\nabla f(x^k)s^k + f(x^k)\right\| \leq \eta^k \left\|f(x^k)\right\|, \text{ for some } \eta^k \in [0, \eta], \tag{11}$$

where $\eta \in [0, 1)$ [19]. This step is not the same as the Newton step and approaches the Newton step as $\eta^k \to 0$. The parameter $\{\eta^k\}$ is referred to as the forcing sequence and determines how accurately the Newton linear equations are solved. Using only the condition (11) with $\eta < 1$ and standard continuous differentiability assumptions on the nonlinear equation to be solved, [19] shows that linear local convergence can be achieved. The additional condition $\eta^k \to 0$ guarantees better than linear (superlinear) local convergence [12, Theor. 11.3]. For η that is sufficiently small, the linear equations $f(x^k)s^k = -f(x^k)$ are solved with progressively smaller error as we get closer to the solution. Iterative linear solvers are especially suited for this since, unlike direct methods, they allow early termination.

For null-space methods, it has been shown in [16] that the problem stays well posed (i.e., the linear matrix coefficient $Z^T F^k Z$ stays strictly positive definite) as long as none of the loops have zero flows in all pipes. Therefore, compared to a Schur method, a null-space algorithm is more robust in dealing with zero flows [13], [18]. However, it is quite usual to have badly conditioned linear systems when large-scale operational networks are considered. For example, Fig. 2 shows the distribution of flow magnitudes, frictional loss coefficients, and the elements of the diagonal matrix $G^k := A_{11}^k$. For the network BWKWnet; see the Appendix for details of networks used in this paper. The ratio of the largest to the smallest friction factors is of order 10^8; this, when coupled with a large range for pipe flows, results in very large condition numbers for the linear system coefficient matrices $Z^T F^k Z$. Therefore, the convergence rate of CG will not be acceptable without approaches that reduce the condition number.

A. Jacobian Regularization as an Inexact Newton Method

To avoid numerical ill conditioning and possible positive semidefiniteness of the linear systems (6) due to zero or very small diagonal elements of A_{11}, zero and very small flows were replaced by an arbitrary small positive number in [15] and related literature; zero flow cases are never allowed for in any link. However, as can be seen in Fig. 2, even when zero flows are replaced by a small constant (for example, by the value 10^{-6}), the condition number of A_{11} is quite large resulting high condition numbers for $Z^T F^k Z$ and $A_{12}^T (F^k)^{-1} A_{12}$ (of the order 10^{12} for this example). A rule of thumb implies a loss of a single decimal place in solving a linear system for every power of 10 increase in the condition number [23].

Therefore, a systematic Jacobian regularization method is proposed in [23] to restrict the condition number of the Newton linear systems (6). Using simple computations, the work in [23] suggests a systematic way to choose a diagonal regularization matrix J so that the condition number of the "regularized" matrix $\tilde{F}^k := F^k + J^k$ is bounded above by some given number $\bar{\kappa}$, that is, $\kappa_2(F^k + J^k) \leq \bar{\kappa}$. For example, choosing

$$J_{jj}^k = \max\left(\frac{\max\limits_{i\in[1,n_p]} F_{ii}^k}{\bar{\kappa}} - F_{jj}^k, 0\right) \tag{12}$$

will guarantee that $\kappa_2(\tilde{F}^k) \leq \bar{\kappa}$. Because $\tilde{F}^k$ is diagonal and invertible, it is straightforward to derive the bound on the 2-norm condition number $\kappa_2(Z^T(\tilde{F}^k)^{-1}Z) \leq \kappa_2(\tilde{F}^k)\kappa(Z)^2$ using the triangle inequality for the matrix norm. This approach reduces the loss of accuracy or convergence caused by solving a badly conditioned system.

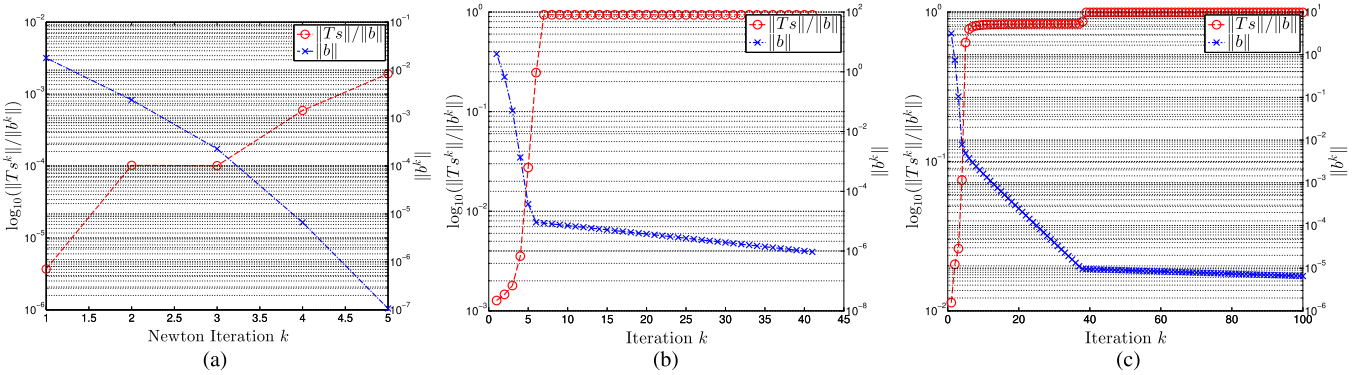


Fig. 3. Plot of $\|T^k\tilde{s}^k\|/\|b^k\|$ and the residual $\|b^k\|$ for the Newton method under different condition number bounds $\bar{\kappa}$. The unregularized Newton method takes 5 steps. BWKWnet network with a HW model. (a) $\bar{\kappa} = 1e5$. (b) $\bar{\kappa} = 1e3$. (c) $\bar{\kappa} = 1e2$.

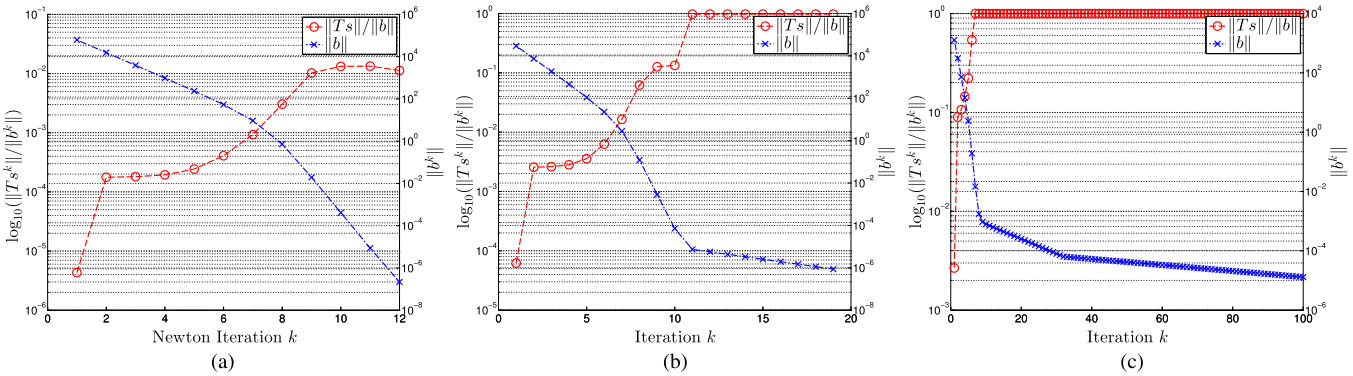


Fig. 4. Plot of $\|T^k\tilde{s}^k\|/\|b^k\|$ and the residual $\|b^k\|$ for the Newton method under different condition number bounds $\bar{\kappa}$. The unregularized Newton method takes 12 steps. Richnet network with a DW model. (a) $\bar{\kappa} = 1e5$. (b) $\bar{\kappa} = 1e4$. (c) $\bar{\kappa} = 1e3$.

Proposition 1: (Regularized Inexact Newton Method) Assume that the Newton method of Algorithm 1 with error tolerance δ_N and inexact linear solvers is modified to regularize the F^k term in (5) with the formula $F^k \leftarrow F^k + J^k$, where J^k is as in (12). Then, an appropriate condition number bound $\bar{\kappa}$ exists such that the regularized scheme is an inexact Newton method, guaranteeing at least linear local convergence.

Proof: Let T^k be the regularization matrix that perturbs the linear system (5) solved at each Newton iterate; that is, $\tilde{A}^k\tilde{s}^k + b^k = 0$ is solved, where $\tilde{A}^k = A^k + T^k$, $A^k := \nabla f(x^k)$, $b^k := f(x^k)$, and $T^k := \mathtt{diag}(J^k, 0)$. Since the matrix T^k is determined only by the condition number constraint $J^k_{jj} = \max((\lambda_{\max}(F^k_{ii})/\bar{\kappa}) - F^k_{jj}, 0)$, its magnitude becomes bigger as we set $\bar{\kappa}$ to be smaller and vice-versa. Now assume this perturbed linear system is solved to a sufficiently small relative error tolerance $e_{tol} \ll 1$ to find the inexact step $\tilde{s}^k$, i.e.,

$$\tilde{A}^k\tilde{s}^k + b^k = \beta^k,\ \|\beta^k\| \le e^k_{tol}\|b^k\|. \tag{13}$$

Substituting for $\tilde{A}^k$ and b^k and rearranging, the residual for the Newton method is $A^k\tilde{s}^k + b = \beta^k - T\tilde{s}$. This implies

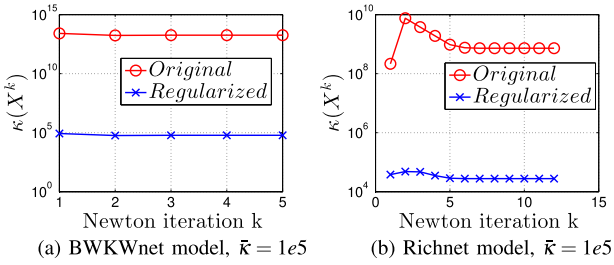
$$\frac{\|A^k\tilde{s}^k + b^k\|}{\|b^k\|} \le \frac{\|\beta^k\|}{\|b^k\|} + \frac{\|T^k\tilde{s}^k\|}{\|b^k\|} \le e^k_{tol} + \frac{\|T^k\tilde{s}^k\|}{\|b^k\|}. \tag{14}$$

For the Newton method with the inexact linear solve process (13), based on the relative error bound (14), we define the "forcing sequence" as

$$\eta^k := e^k_{tol} + \frac{\|T^k\tilde{s}^k\|}{\|b^k\|}. \tag{15}$$

Assume now that e^k_{tol} is sufficiently less than 1 for all k. Considering the nonzero elements of T^k, the ratio $\|T^k\tilde{s}^k\|_\infty/\|b^k\|_\infty$ is bounded by $\max_{j\in[1,n_p]}(\max_{i\in[1,n_p],i\neq j} n_i r_i|q_i|^{n_i-1}/\bar{\kappa}\delta_N)|s^k_j|$. Therefore, for a given tolerance δ_N, the forcing term can be made to satisfy the condition $\eta^k \in [0,1)$ by choosing a sufficiently large bound $\bar{\kappa}$. ■

We have shown here that the forcing term can satisfy the condition $\eta^k \in [0,1)$ provided that the bound on the condition number of the regularized matrix is not too strict. In Figs. 3 and 4, we show how η^k evolves for different bounds on $\bar{\kappa}$ using HW and DW models, respectively. At the early Newton iterates, where $\|b^k\|$ is large, the $\|T^k\tilde{s}^k\|/\|b^k\|$ term is usually small. However, as the condition number of the system increases near convergence, the ratio $\|T^k\tilde{s}^k\|/\|b^k\|$ can grow with increasing magnitudes in J^k and decreasing values for $\|b^k\|$. On the other hand, the decreasing size in the steps s^k diminishes this ratio near convergence. Therefore,



(a) BWKWnet model, $\bar{\kappa} = 1e5$ (b) Richnet model, $\bar{\kappa} = 1e5$

Fig. 5. Typical example of how regularization can reduce the condition number of the matrices $Z^T F^k Z$: (a) and (b) show the evolution of the condition number for the examples in Figs. 3(a) and 4(a), respectively. (a) BWKWnet model $\bar{\kappa} = 1e5$. (b) Richnet model $\bar{\kappa} = 1e5$.

we can keep the $\|T^k \tilde{s}^k\|/\|b^k\|$ term and, therefore, η^k small by making sure $\bar{\kappa}$ is sufficiently large for the given Newton tolerance δ_N.

For the BWKWnet (see the Appendix for details), typical nonregularized condition numbers of $Z^T F^k Z$ can be up to the order $1e^{12}$. Restricting the condition number of F^k to $1e5$ via Jacobian regularization improves the condition number by many orders of magnitude; see Fig. 5(a). A similar plot is shown for the Richnet model in Fig. 5(b). The regularized linear systems result in positive definite reduced systems with better condition numbers, which can be solved more accurately or efficiently by CG. However, if tighter condition number bounds are used, the linear solve solutions will be further away from the Newton steps and so can slow convergence of the Newton iterations as we perturb the linear systems with larger matrices T^k. For $\bar{\kappa} = 1e3$, the forcing sequence η^k approaches 1; we note that little progress is made by the Newton iterates as shown by the convergence properties of the solution errors $\|b^k\|$ in Figs. 3(c) and 4(c). The inexact method is not even able to converge to the required Newton tolerance of $\delta_N = 1e^{-6}$ within the maximum 100 Newton iterations set. After extended time simulations with all network models used in this paper, it seems that setting $\bar{\kappa}$ below about $1e5$ deteriorates convergence. For any network, an appropriate bound can be chosen using test simulations.

V. Tailored Constraint Preconditioners for CG

Subspace iterative methods are an attractive option for solving large-scale sparse saddle-point systems [27]. As outlined in [28], constraint preconditioning can be employed to improve convergence properties in solving the reduced linear systems with any of the Krylov subspace linear solvers, such as CG or MINRES. However, we only investigate preconditioning CG here because, as also discovered for the library of problems in [28], our preliminary analysis showed that preconditioned CG has better performance in solving the positive definite reduced systems (8). The comparison of different Krylov methods is beyond the scope of this paper.

Since we solve the reduced linear system (8) only approximately, it renders the outer iterations into an inexact Newton method. As implied by (15) and the results in Fig. 3, the relative tolerance of the linear solver does not need to be very small for η^k to be uniformly smaller than 1; it need not be much smaller than a small $\|T^k \tilde{s}^k\|/\|b^k\|$ term. This allows the CG solver to terminate early by using a relative error tolerance as an input argument; this ability to tradeoff accuracy and computational time is, of course, not possible with direct solvers. Here, we propose the use of the decreasing function $e^k_{tol,CG} = \min\{\epsilon, \epsilon\|b^k\|\}$, where $\epsilon > 0$ is some sufficiently small number and $\|b^k\|$ is the nonlinear function residual at the kth Newton iterate. Therefore, (8) is solved with a CG tolerance of ϵ at early iterations. As k increases, the tolerance $e^k_{tol,CG}$ becomes proportionally smaller with the Newton residual $\|b^k\|$, finding steps that are closer to the exact Newton search directions at later iterations.

In addition to the required accuracy for the solution, an important property that affects CG performance is the distribution of eigenvalues of the regularized linear systems. Since the convergence properties of iterative schemes worsen when the condition number of the linear systems becomes large, they do not perform well without suitable preconditioners [27]. To solve the null-space reduced system $X^k v = -c^k_z$, where $X^k := Z^T F^k Z$, $-c^k_z$ is the right-hand side of (8), the ideal preconditioner W_z would result in $W_z^{-1} X^k = I$; that is, $W_z = Z^T F^k Z$ results in clustering all of the eigenvalues of the preconditioned system at exactly 1 [29]. However, finding and inverting such a preconditioner at each Newton iterate k is as costly as solving (8).

Therefore, we consider instead preconditioners of the form $W_z := Z^T H Z$, where W_z is SPD, and a constant diagonal matrix H approximates F^k. The proposed preconditioner is called a constraint preconditioner since only the A_{11} block of the saddle-point system is approximated by such a preconditioner, ensuring that the preconditioned system satisfies the linear constraints in (2b) exactly [14], [28]. Algorithm 2 shows preconditioned CG (PCG) applied to the reduced systems in (8). The CG iterations are terminated when the residual norm $\|r_z\|/\|c_z\|$ is sufficiently small or when the maximum number of CG iterations allowed is reached, whichever comes first.

We want to find a tailored preconditioner $W_z = Z^T H^k Z$ that will cluster the eigenvalues of $Z^T F^k Z$ but does not change with Newton iterations, that is, $H^k = H$. For a given network model, this would allow a single sparse incomplete Cholesky factorization [22, Ch. 11] of the preconditioner W_z to be used for all Newton solve CG calls. We note that the spread in the eigenvalues of X^k is due to the varied scales of diagonal elements in F^k matrices and the matrix Z. From Fig. 2 and the formula $F^k_{jj} = n_j r_j |q_j|^{n_j - 1}$ for an HW model, in addition to the large range in flow values, the spread in the different scales in F^k is caused mainly by the spread in the pipe resistance coefficients r_j, which are a constant property of the links. Therefore, we propose the use of the preconditioner $W_1 = Z^T H Z$, where H is the diagonal matrix with $H_{jj} = n_j r_j$. For a DW model, r_j is dependent on the flow conditions and so varies with Newton iterations. However, in laminar and turbulent flows, we have that $r_j \propto L_j D_j^{-4}$ [24, Eq. 2]. Therefore, we propose the use of $H_{jj} = L_j D_j^{-4}$ for the DW case. Alternatively, we consider the simple case where the preconditioner is the orthogonal projection onto the null space $W_2 = Z^T Z$, (i.e., $H = I$) [14].

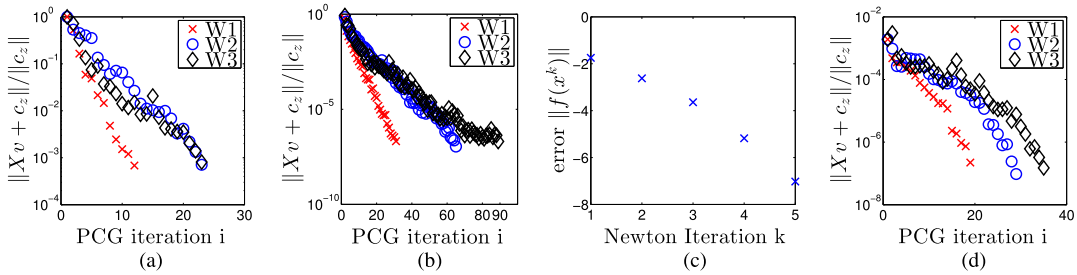


Fig. 6. Comparison of rate of convergence of PCG in solving (8) at Newton iterates k. In (d), warm starting is used at the specified Newton step. Network BWFLnet with HW headloss model used. (a) k = 1. (b) k = 4. (c) Newton iterates k. (d) k = 4, with warm starting.

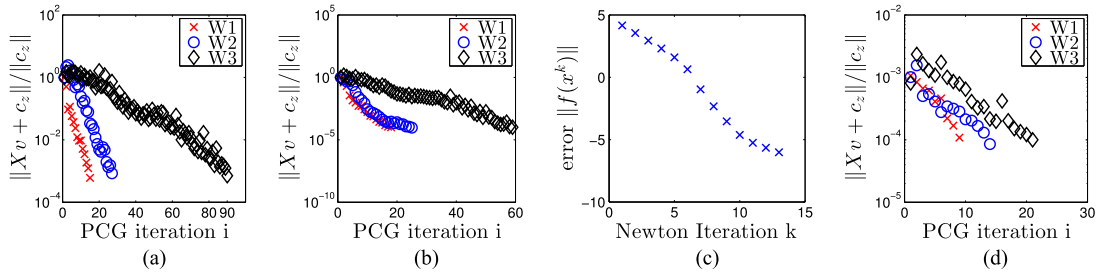


Fig. 7. Comparison of rate of convergence of PCG in solving (8) at Newton iterate. (a) $k = 1$ and (b) $k = 8$. In (d), warm starting is used at the specified Newton step. Network model Wolf Coldera with the DW headloss model used. Preconditioners W1 and W2 are compared with unpreconditioned case W3. (a) k = 1. (b) k = 8. (c) Newton iterates k. (d) k = 8, with warm starting.

Algorithm 2 Preconditioned CG for reduced systems $Z^T F^k Z v + c_z^k = 0$

Initialize:

Choose initial $v \in \mathbb{R}^{n_l}$
Choose a preconditioner W_z
Inputs: F, Z, c_z, v, W_z

$r_z = Z^T F Z v + c_z$, $g_z = W_z \setminus r_z$, $p_z = -g_z$
while Not Converged **do**
  $\alpha \leftarrow r_z^T g_z / p_z^T Z^T F Z p_z$
  $v \leftarrow v + \alpha p_z$
  $r_z^+ \leftarrow r_z + \alpha Z^T F Z p_z$
  $g_z^+ \leftarrow W_z \setminus r_z^+$
  $\beta \leftarrow (r_z^+)^T g_z^+ / (r_z^T g_z)$
  $p_z \leftarrow \beta p_z - g_z^+$
  $r_z \leftarrow r_z^+$, $g_z \leftarrow g_z^+$
end while
return v

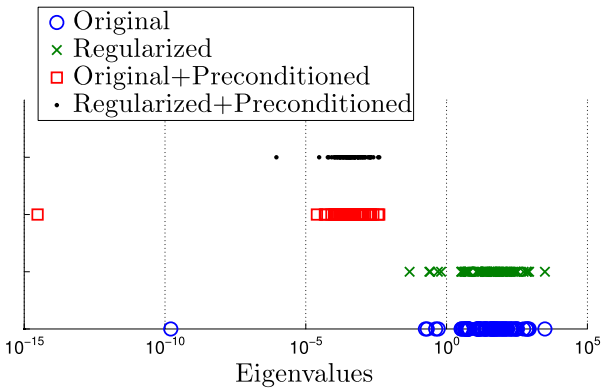
Another factor that affects the convergence properties of CG is the initial condition. For a linear problem $Ax = b$ with solution x^*, the ith CG iterate finds the unique minimizer of the error norm $\|x^i - x^*\|_A$ in the Krylov subspace $\mathscr{K}_i := \text{Span}\{A^j b\}_{j=0}^{j=i}$ and so CG iterates have nonmonotonic convergence of the residual error $\|Ax^i - b\|$. Although solving for x in $Ax = b$ and solving for z in $Az = b - Ax^0$, where $x = z + x^0$ are equivalent, the Krylov subspace iterates generated from the two initial points are not the same. This makes the convergence of CG dependent on the initial guess. The closer the initial guess is to the solution, the fewer iterations CG takes to converge in practice. In our application here, since the coefficient matrix and the right-hand side have similarity to previous Newton iterates, we consider warm-starting each CG call with a solution from a previous Newton iterate.

TABLE I
COMPARISON OF AVERAGE PCG SOLVE TIMES WITH DIFFERENT PRECONDITIONERS VS UNPRECONDIONED SOLVER (W_3). THE W_1 (WS) REFERS TO PRECONDITIONER W_1 WITH WARM STARTING, DATA FROM 500 HYDRAULIC SOLVERS USED

		Speedup: t(W_3) / t(W_i)		
WDN	W_3 (ms)	W_2	W_1	W_1(ws)
CTnet	8.0	1.2	1.9	3.8
Richnet	4.8	1.0	1.2	3.1
WCnet	51.5	2.9	3.6	8.7
BWFLnet	17.1	1.4	2.2	3.0
BWKWnet	23.7	1.2	2.4	4.1

In Figs. 6 and 7, we compare the proposed preconditioners and warm-starting approach using two WDN examples with an HW and a DW headloss model, respectively. In Table I, we test their performance in extended time simulations with 500 demand scenarios in each simulation. All computations were performed within MATLAB R2013b-64 bit for Windows 7.0 installed on a 2.4 GHz Intel Xeon(R) CPU E5-2665 0 with 16 cores. The parameters ϵ, δ_N, and κ were set to 10^{-3}, 10^{-6}, and 10^5, respectively. The details of the water network models can be found in the Appendix.

In Figs. 6(c) and 7(c), we use a single Newton solver (i.e., a single period hydraulic simulation) to show the convergence of the Newton iterations when the CG method is started with a zero initial guess. As the residual of these Newton iterates $\|b^k\| := \|f(x^k)\|$ becomes small, the tolerances passed to the inner CG iterations $\min\{\epsilon, \epsilon\|b^k\|\}$ become smaller. Therefore, as shown in the transition from Fig. 6(a) to (b) and from Fig. 7(a) to (b), the latter Newton iterates will require more



(a) BWKWnet, Preconditioner W_1, $\bar{\kappa} = 1e5$

Fig. 8. Typical case of how regularization and preconditioning change the clustering of eigenvalues for the matrices $Z^T F^k Z$ for the example in Fig. 3(a).

CG iterations. When the solution of previous Newton iterates is used as an initial guess for the CG algorithm, Figs. 6(d) and 7(d) show the number of warm-started CG iterates being reduced by a factor of approximately 2 compared to starting from zero initial conditions for Figs. 6(b) and 7(b), respectively.

In Table I, we show the average CPU times for the PCG solvers in extended time simulations with different preconditioners and warm starting. Since we average the CPU time over large numbers of PCG calls (of order 10^4) for each network model, CPU time can be used as a measure of the number of CG iterations taken. Similar to the results in Figs. 6 and 7, Table I shows the effect of the proposed preconditioners at reducing the number of iterations required for CG convergence. For the HW and DW models, W1 (i.e., the precondioner that takes into account the pipe resistance and null basis) performs the best. For the example networks, compared to the nonpreconditioned case (W3), the preconditioner W1 reduces CG iterates by 2–4 times. The preconditioner W2 also results in significant reductions with numbers of iterations somewhere in between the W1 preconditioned and nonpreconditioned cases. On average, warm starting reduces the computational time of the preconditioned CG by more than a factor of 2. The best PCG implementation, which uses preconditioner W1 with warm starting, reduces computational time by up to a factor of 9 compared to nonpreconditioned CG. In all cases, the matrices are regularized before preconditioning.

Fig. 8 shows a plot of the eigenvalues of the matrices $Z^T F^k Z$, and the effects of Jacobian regularization and preconditioning on the distribution of eigenvalues. Although the eigenvalues are non-negative real for all cases, we plot them one above the other on the Cartesian plane for clarity. Comparing the blue circes with the green x's, we note that regularization does not improve the clustering of eigenvalues; it only moves the eigenvalues close to zero away toward the largest eigenvalue. This is intuitive from (12), where the regularization is designed merely to improve the condition number this way. Preconditioning of the regularized and original (unregularized) matrices results in a tighter clustering of the eigenvalues. In preconditioning the regularized matrix (green x's) with W1, the main distribution of eigenvalues in $[10^{-1}, 10^{4}]$ becomes clustered around the tighter range $[10^{-4}, 10^{-2}]$. This tighter clustering is observed for the unregularized case too. Since better clustering (i.e., tighter clustering of eigenvalues around one or more eigenvalues away from zero) results in improved convergence of iterative solvers, such as CG and MINRES [22], [28], Fig. 8 is in agreement with the results in Table I, where the regularized but unpreconditioned case (W3) has the poorest convergence.

Finally, we employ the best of the PCG linear solvers, to compare the inexact approach with an exact Newton implementation, where the linear systems are solved to an accuracy of 10^{-9} at each Newton iteration. Over all simulations shown in Table I, we sum the total CPU times for the Newton linear solve blocks for comparison. Relative to the exact method, the inexact approach has speed ups of 1.78, 3.23, 2.52, 1.70, and 1.95, respectively, for the networks listed in Table I from top to bottom.

VI. Conclusion

In this paper, we have formulated a null-space inexact Newton method for hydraulic simulation of water distribution networks. The saddle-point structure of the Jacobian in the Newton linear systems is exploited by sparse null-space approaches, which solve the nonlinear hydraulic equations in the kernel space of the mass continuity constraints with less computational resources. In the inexact Newton framework, the nonlinear hydraulic equations are solved to the same level of accuracy as in an exact method but the linear systems for computing the Newton steps are solved only approximately to reduce linear solve times. Iterative linear solvers are used in this framework because they allow early termination with prescribed sufficient accuracy for the linear solvers.

We show that the large variation in flow values, and range of scales in pipe roughness, length, and diameters result in ill-conditioned Newton linear systems. A Jacobian regularization scheme was, therefore, adopted to improve the condition number of the linear systems and to avoid semidefiniteness. By posing the regularized algorithm as an inexact Newton method, we propose practical condition number parameters that keep the regularized system steps not too far away from the Newton steps and so do not require more Newton iterations for convergence.

Since linear solvers CG are not effective without good preconditioners, we have proposed tailored constraint preconditioners. These preconditioners are computationally efficient because they only require a single factorization for multiple solvers as they are functions of only invariant properties of the network model. We have shown that, unlike Jacobian regularization, the new constraint preconditioners result in a much favorable clustering of eigenvalues and so better rates of convergence for the CG linear solver. Numerical experiments with real network and published network models are then used to demonstrate the effectiveness of a preconditioned CG-based inexact Newton method compared to an exact approach. Results show that the new preconditioners, and warm-starting can reduce the CPU time of linear solvers by the CG method by approximately a factor of 4 on average and by up to a factor of 9 for one case. We demonstrate the impact of warm-starting by motivating why it is effective and showing a speed up of about 2

TABLE II
CASE STUDY NETWORKS: deg REPRESENTS THE GRAPH'S AVERAGE DEGREE (I.E., $\deg = 2n_p/n_n$)

Network	n_p	n_n	n_l	n_0	deg	$\frac{nnz(A_{12})}{n_n}$	Headloss Model
CTnet	444	388	48	8	2.24	3.25	HW
Richnet	934	848	86	8	2.20	3.17	DW
WCnet	1976	1770	206	4	2.23	3.22	DW
BWFLnet	2369	2303	66	2	2.05	3.05	HW
BWKWnet	4648	4577	71	1	2.03	3.03	HW

for the linear solvers. We finally compare the proposed inexact Newton method with its exact Newton equivalent and show overall Newton method speed ups of 1.78, 3.23, 2.52, 1.70, and 1.95, respectively, for the example networks and using the best of the proposed constraint preconditioners and warm-starting.

APPENDIX
CASE STUDY NETWORK MODELS

As examples, we use five networks that vary in size and levels of "loopedness," here measured by the average degree of the graph, that is, the average number of pipes incident at each node ($\mathrm{d}eg = 2 * n_p/n_n$). The proprietary operational models BWFLnet and BWKnet [13] are from a typical network in a built up (urban) area in England. The networks Richnet (a real network from Yokshire, U.K.) and WCnet (Wolf-Cordera, part of a real network in Colorado Springs, CO, USA) are detailed in [18]. The relatively smaller size artificial network CTnet is from [30]. Table II summarizes properties of the case study networks and some basic topological characteristics, ordered by increasing size.

REFERENCES

[1] R. Puust, Z. Kapelan, D. Savic, and T. Koppel, "A review of methods for leakage management in pipe networks," *Urban Water J.*, vol. 7, no. 1, pp. 25–45, 2010.

[2] A. B. A. Sakarya and L. W. Mays, "Optimal operation of water distribution pumps considering water quality," *J. Water Resource Planning Manage.*, vol. 126, no. 4, pp. 210–220, 2000.

[3] P. W. Jowitt and G. Germanopoulos, "Optimal pump scheduling in water-supply networks," *J. Water Resource Planning Manage.*, vol. 118, no. 4, pp. 406–422, 1992.

[4] M. Allen, A. Preis, M. Iqbal, and A. Whittle, "Water distribution system monitoring and decision support using a wireless sensor network," in *Proc. 14th ACIS Int. Conf. Softw. Eng., Artif. Intell., Netw. Parallel/Distrib. Comput.*, Jul. 2013, pp. 641–646.

[5] P. Belotti, C. Kirches, S. Leyffer, J. Linderoth, J. Luedtke, and A. Mahajan, "Mixed-integer nonlinear optimization," *Acta Numerica*, vol. 22, pp. 1–131, 2013.

[6] H. R. Maier *et al.*, "Evolutionary algorithms and other metaheuristics in water resources: Current status, research challenges and future directions," *Environ. Model. Softw.*, vol. 62, no. 0, pp. 271–299, 2014.

[7] EPANET 2: Users Manual L. A. Rossman, Ed., 2000.

[8] D. A. Savic and G. A. Walters, "Genetic algorithms for least-cost design of water distribution networks," *J. Water Resource Plan. Manage.*, vol. 123, no. 2, pp. 67–77, 1997.

[9] B. Ulanicki, J. Kahler, and H. See, "Dynamic optimization approach for solving an optimal scheduling problem in water distribution systems," *J. Water Resource Plan. Manage.*, vol. 133, no. 1, pp. 23–32, 2007.

[10] A. Drud, "CONOPT: A GRG code for large sparse dynamic nonlinear optimization problems," *Math. Program.*, vol. 31, no. 2, pp. 153–191, 1985.

[11] R. Wright, E. Abraham, P. Parpas, and I. Stoianov, "Control of water distribution networks with dynamic DMA topology using strictly feasible sequential convex programming," *Water Resources Res.*, vol. 51, no. 12, pp. 9925–9941, 2015, doi: 10.1002/2015WR017466.

[12] J. Nocedal and S. J. Wright, *Numerical Optimization*. Berlin, Germany: Springer-Verlag, 2006.

[13] E. Abraham and I. Stoianov, "Sparse null space algorithms for hydraulic analysis of large scale water supply networks," *J. Hydraul. Eng.*, vol. 142, no. 3, 2016, Art. no. 4015058. doi:10.1061/(ASCE)HY.1943-7900.0001089.

[14] N. Gould, M. Hribar, and J. Nocedal, "On the solution of equality constrained quadratic programming problems arising in optimization," *SIAM J. Scientif. Comput.*, vol. 23, no. 4, pp. 1376–1395, 2001.

[15] E. Todini and L. A. Rossman, "Unified framework for deriving simultaneous equation algorithms for water distribution networks," *J. Hydraul. Eng.*, vol. 139, no. 5, pp. 511–526, 2012.

[16] H. B. Nielsen, "Methods for analyzing pipe networks," *J. Hydraul. Eng.*, vol. 115, no. 2, pp. 139–157, 1989.

[17] H. Rahal, "A co-tree flows formulation for steady state in water distribution networks," *Adv. Eng. Softw.*, vol. 22, no. 3, pp. 169–178, 1995.

[18] S. Elhay, A. R. Simpson, J. Deuerlein, B. Alexander, and W. Schilders, "A reformulated co-tree flows method competitive with the global gradient algorithm for solving the water distribution system equations," *J. Water Resources Plan. Manage.*, 2014.

[19] R. S. Dembo, S. C. Eisenstat, and T. Steihaug, "Inexact newton methods," *SIAM J. Numer. Anal.*, vol. 19, no. 2, pp. 400–408, 1982.

[20] A. Shahzad, E. C. Kerrigan, and G. A. Constantinides, "A stable and efficient method for solving a convex quadratic program with application to optimal control," *SIAM J. Optimiz.*, vol. 22, no. 4, pp. 1369–1393, 2012.

[21] Y. Saad, *Iterative Methods for Sparse Linear Systems*, 2nd ed. Philadelphia, PA: SIAM, 2003.

[22] A. Greenbaum, *Iterative Methods for Solving Linear Systems*. Philadelphia, PA, USA: SIAM, 1997, vol. 17.

[23] S. Elhay and A. R. Simpson, "Dealing with zero flows in solving the nonlinear equations for water distribution systems," *J. Hydraul. Eng.*, vol. 137, no. 10, pp. 1216–1224, 2011.

[24] A. Simpson and S. Elhay, "Jacobian matrix for solving water distribution system equations with the darcy-weisbach head-loss model," *J. Hydraul. Eng.*, vol. 137, no. 6, pp. 696–700, 2010.

[25] J. E. Dennis Jr. and R. B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. Philadelphia, PA, USA: SIAM, 1996, vol. 16.

[26] R. Epp and A. G. Fowler, "Efficient code for steady-state flows in networks," *J. Hydraul. Div.*, vol. 96, no. 1, pp. 43–56, 1970.

[27] M. Benzi, G. H. Golub, and J. Liesen, "Numerical solution of saddle point problems," *Acta Numer.*, vol. 14, pp. 1–137, 2005.

[28] C. Keller, N. I. Gould, and A. J. Wathen, "Constraint preconditioning for indefinite linear systems," *SIAM J. Matrix Anal. Appl.*, vol. 21, no. 4, pp. 1300–1317, 2000.

[29] N. Gould, D. Orban, and T. Rees, "Projected krylov methods for saddle-point systems," *SIAM J. Matrix Anal. Appl.*, vol. 35, no. 4, pp. 1329–1343, 2014.

[30] A. Ostfeld *et al.*, "Battle of the water calibration networks," *J. Water Resources Plan. Manage.*, vol. 138, no. 5, pp. 523–532, 2011.

Edo Abraham (M'17) received the M.Eng degree in electrical and electronic engineering and the PhD degree in control engineering from Imperial College London, London, U.K., in 2008 and 2013, respectively.

He is currently a Research Associate in the Environmental and Water Resources Engineering Group, Imperial College London. His research interests focus on the development of optimal control and robust estimation methods, graph-theoretic methods for the analysis of network systems, and tailored nonlinear programming tools for the optimal operation and design of smart water distribution networks.

Ivan Stoianov received the M.Sc. and Ph.D. degrees in civil and environmental engineering from Imperial College, London, U.K., in 1997 and 2003, respectively.

Currently, he is a Senior Lecturer in Water Systems Engineering in the Department of Civil and Environmental Engineering, Imperial College. He has founded and leads InfraSense Labs at Imperial College London, U.K. His research interests include experimental and analytical studies in hydraulics, hydroinformatics, and operational management of water supply systems.